Degree-constrained Subgraph Reconfiguration is in P

Moritz Mühlenthaler*

August 7, 2015

Abstract

The degree-constrained subgraph problem asks for a subgraph of a given graph such that the degree of each vertex is within some specified bounds. We study the following reconfiguration variant of this problem: Given two solutions to a degree-constrained subgraph instance, can we transform one solution into the other by adding and removing individual edges, such that each intermediate subgraph satisfies the degree constraints and contains at least a certain minimum number of edges? This problem is a generalization of the matching reconfiguration problem, which is known to be in P. We show that even in the more general setting the reconfiguration problem is in P.

1 Introduction

A reconfiguration problem asks whether a given solution to a combinatorial problem can be transformed into another given solution in a step-by-step fashion such that each intermediate solution is "proper", where the definition of proper depends on the problem at hand. For instance, in the context of vertex coloring reconfiguration, "step-by-step" typically means that the color of a single vertex is changed at a time, and "proper" has the usual meaning in the graph coloring context. An issue of particular interest is the relation between the complexity of the underlying combinatorial problem and its reconfiguration variant. This complexity relation has been studied for classical combinatorial problems including for example graph coloring, satisfiability, matching, and the shortest path problem [4, 1, 2]. Surprisingly, the reconfiguration variants of some tractable problems turn out to be intractable [4], and vice versa [7]. An overview of recent results on reconfiguration problems can be found in [6].

In this work we investigate the complexity of the reconfiguration problem associated with the (a, b)-degree-constrained subgraph (ab-DCS) problem. Let $G = (V, E)$ be a graph and let $a, b : V \to \mathbb{N}$ be two functions called *degree bounds* such that for each vertex v of G we have $0 \leq a(v) \leq b(v) \leq \delta_G(v)$, where $\delta_G(v)$ denotes the degree of v in G. The task is to decide if there is a subgraph S of G that satisfies the degree constraints, that is, for each vertex v of S, $\delta_S(v)$ is required to be at least $a(v)$ and at most $b(v)$. Typically, the intention is to find among all subgraphs of G that satisfy the degree constraints one with the greatest number of edges. This problem is a generalization of the classical maximum matching problem and can be solved in polynomial time by a combinatorial algorithm [5, 3]. The ab-DCS reconfiguration problem is defined as follows:

Definition 1. *(st-DCSConn)*
INSTANCE: *An ab-DCS instance, source and target solutions M, N, an integer $k \geq 1$.*
QUESTION: *Is it possible to transform M into N by adding/removing a single edge in each step such that each intermediate subgraph satisfies the degree constraints and contains at least* $\min\{|E(M)|, |E(N)|\} - k$ *edges?*

Our main result is the following

Theorem 1. st-DCSConn *can be solved in polynomial time.*

*Research funded in parts by the School of Engineering of the University of Erlangen-Nuremberg.

It was shown by Ito et al. in [4, Proposition 2] that the analogous matching reconfiguration problem can be solved in polynomial time. According to our result, the reconfiguration problem remains tractable even in the more general ab-DCS reconfiguration setting. The proof of the main result essentially contains an algorithm that determines a suitable sequence of edge additions/removals if one exists. The number of reconfiguration steps is bounded by $O(|E|^2)$. The algorithm also provides a certificate for No-instances.

2 Notation

In this paper we deal with subgraphs of some simple graph $G = (V, E)$, which is provided by ab-DCS a problem instance. The subgraphs of concern are induced by subsets of E. For notational convenience, we identify these subgraphs with the subsets of E and can therefore use standard set-theoretic notation ($\cap$, $\cup$, $-$) for binary operations on the subgraphs. Let H and K be two subgraphs of G, denoted by $H, K \subseteq G$. By $E(H)$ we refer explicitly to the set of edges of the graph H. We write $H + K$ for the union of H and K if they are disjoint. By $H \triangle K$ we denote the *symmetric difference* of H and K, that is $H \triangle K := (H - K) + (K - H)$. If e is an edge of G we may write $H + e$ and $H - e$ as shorthands for $H + \{e\}$ and $H - \{e\}$, respectively. We denote the degree of a vertex v of H by $\delta_H(v)$. A walk $v_0 \overset{e_0}{—} \dots \overset{e_{t-1}}{—} v_t$ in G is a *trail* if $e_0, \dots, e_{t-1}$ are distinct. The vertices v_0 and v_t are called *end vertices*, all other vertices are called *interior*. A trail without an edge is called *empty*. A non-empty trail is *closed* if its end vertices agree, otherwise it is *open*. A closed trail is also called a *cycle*. In a slight abuse of notation we will sometimes consider trails in G simply as subgraphs of G and combine them with other subgraphs using the notation introduced above. A trail is called (K, H)*-alternating* if its edges, in the order given by the trail, are alternatingly chosen from $K - H$ and $H - K$. An odd-length (K, H)-alternating trail T is called K-augmenting if $|E(K \triangle T)| > |E(K)|$.

Let G be the graph and $a, b : V \to \mathbb{N}$ be the degree bounds of an ab-DCS instance. A subgraph $M \subseteq G$ that satisfies the degree constraints is called *ab-constrained*. A vertex v of M is called *a-tight* (*b-tight*) *in* M if $\delta_M(v) = a(v)$ ($\delta_M(v) = b(v)$). A vertex is called *ab-fixed in* M if it is both a-tight and b-tight in M. We say that M is a-tight (b-tight) if each vertex of M is a-tight (b-tight). A closed (M, N)-alternating trail $T = v_0 — \dots — v_t$ of even length is called *alternatingly ab-tight* in M if for each i, $0 \le i \le t$, v_i is a-tight iff i is even and b-tight iff i is odd, or vice versa.

3 ab-Constrained Subgraph Reconfiguration

Throughout this section, we assume that we are given some st-DCSConn instance (G, M, N, a, b, k), where $G = (V, E)$ is a graph, $a, b : V \to \mathbb{N}$ are degree bounds, $M, N \subseteq G$ are ab-constrained, and $k \ge 1$. A *reconfiguration step*, or *move*, adds/removes an edge to/from a subgraph. Given an M-edge e and an N-edge e', an *elementary move* on M yields $M - e + e'$, either by adding e' after removing e or vice versa. M is *k-reconfigurable* to N if there is a sequence of reconfiguration moves that transforms M into N such that each intermediate subgraph respects the degree constraints and contains at least $\min\{|E(M)|, |E(N)|\} - k$ edges. Clearly, if M is k-reconfigurable to N then M is also k'-reconfigurable to N for any $k' > k$. M is *internally k-reconfigurable* to N if it is k-reconfigurable under the additional restriction that each intermediate subgraph is contained in $M \triangle N$. If M is not internally k-reconfigurable to N but still k-reconfigurable to N then we say that M is externally k-reconfigurable to N.

The general procedure for deciding if M is k-reconfigurable to N is the following: First, we check for the presence of obstructions that render a reconfiguration impossible. If it turns out that reconfiguration is still possible we reconfigure (M, N)-alternating trails in $M \triangle N$, one by one, until we either finish successfully or we obtain a certificate for M not being k-reconfigurable to N. Curiously, it turns out that if M is not 2-reconfigurable to N then M is not k-reconfigurable to N for any $k \ge 2$.

3.1 Obstructions

When transforming M into N, certain parts of M may be "fixed" and therefore make a proper reconfiguration impossible. Similar obstructions occur for example in vertex coloring reconfiguration, where certain vertices are fixed in the sense that their color cannot be changed [2]. In our case we identify a certain subgraph of G which depends on M and N and cannot be changed at all.

Let v be an ab-fixed vertex of G. When reconfiguring M to N, no M-edge incident to an ab-fixed vertex can be removed and no $(G - M)$-edge incident to an ab-fixed vertex can be added during a reconfiguration process without violating the degree constraints. Hence an edge is fixed if it is incident to an ab-fixed vertex. However, we may identify larger parts of G that are fixed due to the given subgraph M and the degree bounds. If we consider each ab-fixed edge to be M-fixed, then we can identify further M-fixed edges based on the following observations: First if a vertex v is incident to exactly $b(v)$ M-edges and each of them is M-fixed then all edges incident to v are M-fixed. Similarly, if v is incident to exactly $a(v)$ M-edges and each $(G - M)$-edge incident to v is M-fixed then each edge incident to v is M-fixed. Algorithm 1 shows how to identify an M-fixed subgraph of G based on these observations. By $\mathrm{I}_M(v)$ we denote the set of M-edges incident to the vertex v. Some bookkeeping could be employed to speed things up, but it is not necessary for our argument.

Algorithm 1: M-FixedSubgraph

```
input  : $ab$-DCS instance $(G, a, b)$, $ab$-constrained $M \subseteq G$
output: $M$-fixed subgraph $F \subseteq G$
$F \longleftarrow \emptyset$;      $F' \longleftarrow \{e \in G \mid e \text{ is fixed}\}$
while  $|F'| > |F|$ do
    $F \longleftarrow F'$
    if $\delta_M(v) = b(v)$ and $\mathrm{I}_M(v) \subseteq F$ then
        $F' \longleftarrow F' \cup \mathrm{I}_G(v)$
    if $\delta_M(v) = a(v)$ and $\mathrm{I}_{G-M}(v) \subseteq F$ then
        $F' \longleftarrow F' \cup \mathrm{I}_G(v)$

end
return $F$
```

Proposition 1. *Let M, N be ab-constrained subgraphs of G and let $F \subseteq G$ be M-fixed. If $(M \bigtriangleup N) \cap F$ is non-empty then M is not k-reconfigurable to N for any $k \geq 1$.*

That is, any M-fixed edge in $(M \bigtriangleup N)$ is a No-certificate. As a consequence, we can check if M and N agree on the subgraph $F \subseteq G$ found by Algorithm 1 as a preprocessing step. At this point in particular, but also later on it will be convenient to consider *subinstances* of a given st-DCSConn instance $\mathcal{I} = (G, M, N, a, b, k)$. If $H \subseteq G$ then the corresponding subinstance $\mathcal{I}_H$ is the instance $(H, M \cap H, N \cap H, a_H, b_H, k)$, where

$$\begin{aligned} a_H(v) &= \max\{0, a(v) - \delta_{(G-H)\cap M}(v)\} \\ b_H(v) &= b(v) - \delta_{(G-H)\cap M}(v) \ . \end{aligned}$$

Proposition 2. *If $(M \bigtriangleup N) \cap F = \emptyset$ then $\mathcal{I}$ is a Yes-instance if and only if $\mathcal{I}_{G-F}$ is a Yes instance.*

The graph $G - F$ does not have any fixed vertices and hence the $(M - F)$-fixed subgraph produced by Algorithm 1 is empty. Removing the M-fixed subgraph of G in a preprocessing step will considerably simplify our arguments later on. It should be immediate that no fixed edges can be introduced by reconfiguring an ab-constrained subgraph.

3.2 Internal Alternating Trail Reconfiguration

The next Lemma is our fundamental tool for reconfiguring alternating trails in $M \bigtriangleup N$. For any such trail T, it provides necessary and sufficient conditions for $T \cap M$ being internally 1-reconfigurable to $T \cap N$ by performing only elementary moves. Behind the scenes, T is recursively divided into subtrails which need to be reconfigured in a certain order. If successful, the reconfiguration procedure performs exactly $|E(T)|$ edge additions/removals.

Lemma 1. *Let $T = v_0 \overset{e_0}{\text{---}} \dots \overset{e_{t-1}}{\text{---}} v_t$ be a (M, N)-alternating trail of even length in $M \bigtriangleup N$. Then $T \cap M$ is internally 1-reconfigurable to $T \cap N$ using only elementary moves if and only if T it satisfies each of the following conditions:*

1. *T contains no ab-fixed vertex.*
2. *If T is open then v_0 is not a-tight and v_t is not b-tight in M.*
3. *If T is closed then each of the following is true:*
   (a) *T is not b-tight in M*
   (b) *T is not a-tight in M*
   (c) *T is not alternatingly ab-tight in M*

Proof. Without loss of generality, let e_0 be an M-edge. We first show the necessity of conditions 1–3. By Proposition 1, if T contains an M-fixed vertex then $T \cap M$ is not k-reconfigurable to $T \cap N$ for any $k \geq 1$. If T is open and v_0 is a-tight in M, then e_0 cannot be removed from $T \cap M$ without violating the degree constraints. Likewise, if T is open and v_t is b-tight then e_t cannot be added to $T \cap M$. If T is closed and b-tight in M, i.e., 3a is violated, then no N-edge can be added after removing any M-edge. Similarly, if T is closed and a-tight in M, i.e., 3b is violated, then no M-edge can be removed after adding a single N-edge. If T is closed and alternatingly ab-tight then no edge can be added to or removed from $T \cap M$, so it cannot be internally 1-reconfigurable to $T \cap N$. In summary, if any of the conditions 1–3 is violated then $T \cap M$ is not internally 1-reconfigurable to $T \cap N$ using elementary moves.

In order to show the sufficiency of conditions 1–3 we employ the following general strategy: We partition T into (M, N)-alternating subtrails R, Q, S, each of even length and at least two of them non-empty. We show that for an appropriate choice of these subtrails there is an ordering, say Q, R, S, such that the first two subtrails are non-empty and Q satisfies conditions 1–3 in M, R satisfies the same conditions in $M - (Q \cap M) + (Q \cap N)$, and S, if non-empty, in turn satisfies the conditions in $M - (Q \cap M) + (Q \cap N) - (R \cap M) + (R \cap N)$. Therefore, each non-empty subtrail can be dealt with individually in a recursive fashion as long as the ordering is respected. The base case of the recursion consists of an (M, N)-alternating trail $B = u — v — w$ of length two. We show that if B satisfies 1–3 then $B \cap M$ is internally 1-reconfigurable to $B \cap N$ by an elementary move. Since B satisfies conditions 1 and 2, u is not a-tight and w is not b-tight. If v is b-tight we can remove $u — v$ from M and add $v — w$ to $M - (u — v)$ without violating the degree constraints in any step. Similarly, if v is not b-tight we can add $v — w$ to M and afterwards remove $u — v$ from $M + (v — w)$ without violating the degree constraints. So $B \cap M$ is internally 1-reconfigurable to $B \cap N$ by an elementary move as required. For the general recursion, we consider two main cases: T is either open or closed.

We first assume that T is open. Since T satisfies conditions 1 and 2, there is some i, $0 \leq i < t - 1$, i even, such that v_i is not a-tight and v_{i+2} is not b-tight in M. To see this, assume that there is no such i. Then, by induction, for each $0 \leq i < t - 1$, i even, v_{i+2} must b-tight because v_i is not a-tight. However, by condition 2, v_t is not b-tight in M, a contradiction. We pick R, Q, and S as follows

$$\underbrace{v_0 \overset{e_0}{—} v_1 \overset{e_1}{—} \dots \overset{e_{i-1}}{—} v_i}_{R}, \quad \underbrace{v_i \overset{e_i}{—} v_{i+1} \overset{e_{i+1}}{—} v_{i+2}}_{Q}, \quad \underbrace{v_{i+2} \overset{e_{i+2}}{—} v_{i+3} \overset{e_{i+3}}{—} \dots \overset{e_{t-1}}{—} v_t}_{S} .$$

Note that Q is an open trail satisfying conditions 1–3 and R, S, if non-empty, can be open or closed. At this point $Q \cap M$ is internally 1-reconfigurable to $Q \cap N$ as described above, and the result is $M - e_i + e_{i+1}$. It is readily verified that if R and S are open then they satisfy conditions 1–3 in $M - e_i + e_{i+1}$ and can therefore be treated independently after reconfiguring Q. However, at least one of R, S being closed leads to a slight complication.

Let us assume that S is closed. Note that, by assumption, v_{i+2} is not b-tight in M and therefore it cannot be a-tight in $M - e_i + e_{i+1}$. Thus, if any of the conditions 1–3 is violated in M then it cannot be violated in $M - e_i + e_{i+1}$. Since 3a–3c cannot be violated simultaneously we conclude that S satisfies conditions 1–3 either in M or in $M - e_i + e_{i+1}$. An analogous argument shows that R satisfies conditions 1–3 either in M or in $M - e_i + e_{i+1}$. Therefore, there is an ordering of R, Q, S that is consistent with the general strategy outlined above and depends on the tightness of the vertices v_i and v_{i+2} in M. As a visual aid, Figure 1 shows a example of an open (M, N)-alternating trail of even length, where a proper choice of R, Q, and S causes R and S to be both closed. In the shown example the solid edges belong to M and the dashed edges to N.

It remains to be shown that if T is closed and T satisfies properties 1–3 then $T \cap M$ is internally 1-reconfigurable to $T \cap N$ using only elementary moves. For this purpose we find a partition of T

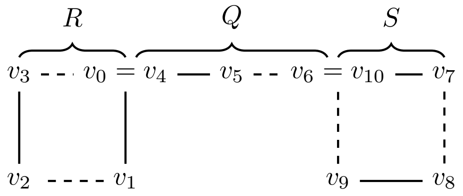

Figure 1: Decomposition of an (M, N)-alternating trail $T = v_0 — \ldots — v_{10}$ of even length into three trails R, Q, and S. Note that T is open, but R and S are closed.

into subtrails that is compatible with our general strategy above. In particular, we show that if T is closed and satisfies properties 1–3 then T can be partitioned into two non-empty open trails R and Q, both of even length, such that Q satisfies 1–3. Furthermore, we show that R satisfies conditions 1–3 in $M - (Q \cap M) + (Q \cap N)$. That is, R can be dealt with after reconfiguring $Q \cap M$ to $Q \cap N$. We pick any two vertices of T that are connected by an M-edge, say v_0 and v_1, and consider the following cases:

(i) v_0 is neither a-tight nor b-tight, or v_1 is neither a-tight nor b-tight

(ii) v_0 is a-tight and v_1 is b-tight, or v_0 is b-tight and v_1 is a-tight

(iii) v_0 and v_1 are both b-tight

(iv) v_0 and v_1 are both a-tight

Case (i) We assume without loss of generality, that v_0 is neither a-tight nor b-tight, since if v_0 is a-tight or b-tight, then v_1 must be neither a-tight nor b-tight and if this is the case we can rearrange the vertices of T in the following way

$$T = v_1 — v_0 — v_{t-1} — \ldots — v_2 — v_1 \tag{1}$$

and establish that v_0 is neither a-tight nor b-tight. Now we choose R and Q as follows:

$$R = v_2 — v_3 — \ldots — v_t, \quad Q = v_0 — v_1 — v_2$$

If v_2 is not b-tight, then Q satisfies 1–3 and $Q \cap M$ can be reconfigured instantly to $Q \cap N$ by an elementary move. That is, we obtain $M - e_0 + e_1$ without violating the degree constraints. Now v_0 cannot be b-tight and v_2 cannot be a-tight in $M - e_0 + e_1$. Therefore, R now satisfies 1–3 and can be reconfigured as shown in the first main case of the proof. If v_2 is b-tight in M, then, by analogous considerations, R satisfies 1–3 and Q satisfies conditions 1–3 in $M - (R \cap M) + (R \cap N)$.

Case (ii) Without loss of generality, we assume that v_0 is a-tight and v_1 is b-tight, since if not, we can rearrange the vertices of T according to Eq. (1). Due to property 3c, T is not alternatingly ab-tight, so there is some i, $0 \leq i < t$, i even, such that v_i is not a-tight or v_{i+1} is not b-tight. If v_i is not a-tight, we choose R and Q to be

$$Q = v_i — v_{i+1} — \ldots — v_t (= v_0), \quad R = v_0 — v_1 — \ldots — v_i$$

Otherwise, v_{i+1} is not b-tight and we pick R and Q as follows

$$Q = v_1 — v_0 — \ldots — v_{i+1}, \quad R = v_{i+1} — v_i — \ldots — v_1$$

Either way, Q satisfies satisfies conditions 1–3 in M and R satisfies the same conditions in $M - (Q \cap M) + (Q \cap N)$.

Case (iii) If v_0 and v_1 are both b-tight then, by property 3a, there is some i, $0 \leq i < t$, such that v_i is not b-tight. Without loss of generality, we assume that i is even. We pick Q and R as follows:

$$Q = v_0 — v_1 — \ldots — v_i, \quad R = v_i — v_{i-1} — \ldots — v_0$$

Then Q satisfies 1–3 and R satisfies the same conditions in $M - (Q \cap M) + (Q \cap N)$.

Case (iv) This case is analogous to case (iii). By property 3b, there is some i, $0 \leq i < t$, such that v_i is not a-tight. Again, without loss of generality, we assume that i is even. We pick Q and R as follows

$$Q = v_0 \text{ — } v_1 \text{ — } \dots \text{ — } v_i, \quad R = v_i \text{ — } v_{i-1} \text{ — } \dots \text{ — } v_0$$

and conclude that Q satisfies 1–3 in M and R satisfies 1–3 in $M - (Q \cap M) + (Q \cap N)$.

From our consideration of the various cases we conclude that a given (M, N)-alternating trail T that satisfies conditions 1–3 can be recursively partitioned into subtrails as outlined in the general strategy above. Since the single base case of the recursion employs only elementary moves on edges of T, $T \cap M$ is internally 1-reconfigurable to $T \cap N$ using only elementary moves. □

An (M, N)-alternating trail is *maximal* if there is no suitable edge in $M \triangle N$ to extend the trail at one of its end nodes. The subsequent lemmas establish sufficient conditions for maximal alternating trails to be internally 1- or 2-reconfigurable. Such trails are important in the proof of Theorem 1. Note however, that in contrast to Lemma 1 we are not restricted to elementary moves. In the following, let $T = v_0 \overset{e_0}{\text{—}} \dots \overset{e_{t-1}}{\text{—}} v_t$ be a maximal (M, N)-alternating trail in $M \triangle N$ such that no vertex of T is ab-fixed.

Lemma 2. *If T is open and has even length then $T \cap M$ is internally 1-reconfigurable to $T \cap N$.*

Proof. Without loss of generality, we assume that $e_0 \in M$. Then $\delta_M(v_0) > \delta_N(v_0)$ and v_0 is not a-tight in M. Since T is maximal and has even length, $e_{t-1} \in N$. Therefore, $\delta_M(v_t) < \delta_N(v_t)$ and thus v_t is not b-tight in M. By assumption, T has no ab-fixed vertex. Therefore, $T \cap M$ is internally 1-reconfigurable to $T \cap N$ by Lemma 1. □

Lemma 3. *If T has odd length and e_0 is an N-edge then*

1. *$T \cap M$ is internally 1-reconfigurable to $T \cap N$, and*
2. *$T \cap N$ is internally 2-reconfigurable to $T \cap M$.*

Proof. We first prove part 1 of the statement. Since T is maximal, the end nodes v_0 and v_t are not b-tight in M. We recursively divide T into two subtrails R and S, such that R has even length and S has odd length and no end vertex of S is b-tight, either in M or in $M - (R \cap M) + (R \cap N)$. The base case of this recursion that is not covered by Lemma 1 consists of S having a single N-edge. Since none of the end vertices is b-tight, the remaining N-edge can be added without violating the degree constraints.

Two subcases that occur when dividing T into subtrails R and S. First assume that there is some i, $0 < i < t$, such that v_i is not a-tight. If i is even then we pick $R = v_i \text{ — } \dots \text{ — } v_0$ and choose $S = v_i \text{ — } \dots \text{ — } v_t$. If i is odd, we pick $R = v_i \text{ — } \dots \text{ — } v_t$ and $S = v_0 \text{ — } \dots \text{ — } v_i$. Either way, $R \cap M$ is internally 1-reconfigurable to $R \cap N$ by Lemma 1. Furthermore, S is an odd-length (N, M)-alternating and no end vertex of S is b-tight in $M - (R \cap M) + (R \cap N)$. On the other hand, if there is no such i, then we pick $R = v_1 \text{ — } \dots \text{ — } v_t$ and $S = v_0 \overset{e_0}{\text{—}} v_1$. Now, $M + e_0$ does not violate the degree constraints and $R \cap (M + e_0)$ is internally 1-reconfigurable to $R \cap (N - e_0)$ according to Lemma 1, since v_1 is not a-tight in $M + e_0$ and v_t is not b-tight by assumption. Therefore, in each case the subtrails are 1-reconfigurable as required.

The proof of part 2 is somewhat analogous: Since T is maximal, v_0 and v_t cannot be a-tight in N. We again recursively divide T into two subtrails R and S, such that R has even length and S has odd length and no end vertex of S is a-tight, in M or in $M - (R \cap M) + (R \cap N)$. We distinguish the following two cases: First, we assume that each internal vertex of T is b-tight. Then we pick $S = v_0 \text{ — } v_1$ and $R = v_1 \text{ — } \dots \text{ — } v_t$. We can reconfigure S by removing e_0 and note that R is internally 1-reconfigurable in $(M - e_{t-1}) \triangle N$ by Lemma 1. Therefore, T is 2-reconfigurable. Now, assume that there is some i, $0 < i < t$, such that v_i is not b-tight. If i is even we choose $R = v_0 \text{ — } \dots \text{ — } v_i$ and $S = v_t \text{ — } \dots \text{ — } v_{i+1}$. If i is odd we swap the choices of R and S. Now, $R \cap M$ is internally 1-reconfigurable to $R \cap N$ and no end vertex of S is a-tight, which means we can proceed recursively by reconfiguring S. □

Lemma 4. *If T is closed and has even length then*

1. *$T \cap M$ is internally 2-reconfigurable to $T \cap N$ if T is not alternatingly ab-tight, and*
2. *$T \cap M$ is internally 1-reconfigurable to $T \cap N$ if T is neither b-tight nor alternatingly ab-tight.*

Proof. If the conditions 3a– 3c of Lemma 1 are satisfied then we get that $T \cap M$ is internally 1-reconfigurable to $T \cap N$. Thus, to complete the proof it is sufficient to consider the case that one of these conditions is violated. In order to prove statement 1 assume that T is alternatingly ab-tight, i.e., 3c is violated. Then $T \cap M$ is not internally k-reconfigurable to $T \cap N$ for any $k \geq 1$, since no edges of $T \cap M$ can be removed and no edges of $T \cap N$ can be added without violating the degree constraints. If T is b-tight, i.e., 3a is violated, then we choose $R = v_1 \text{ — } \ldots \text{ — } v_{t-1}$ and after removing the M-edge e_0, $R \cap M$ is internally 1-reconfigurable to $R \cap N$ by Lemma 1. Therefore, $T \cap M$ is internally 2-reconfigurable to $T \cap N$. Now, suppose that T is a-tight (3b is violated), then we can add the N-edge e_{t-1} and choose $R = v_0 \text{ — } \ldots \text{ — } v_{t-1}$. Then, by Lemma 1, $R \cap M$ is internally 1-reconfigurable to $R \cap N$, which completes the proof of statement 1.

In order to prove statement 2 it is sufficient to reconsider the case that T is b-tight. If this is the case then no N-edge can be added after removing any M-edge and therefore, $T \cap M$ is not internally 1-reconfigurable to $T \cap N$. □

3.3 External Alternating Trail Reconfiguration

In the following we deal with even-length alternating cycles that are either alternatingly ab-tight or b-tight. The two cases are somewhat special since we will need to consider edges that are not part of the cycles themselves. Let $C = u_0 \overset{e_0}{\text{—}} \ldots \overset{e_{t-1}}{\text{—}} u_t$ be an (M, N)-alternating cycle of even length in $M \triangle N$. We will first consider the case that C is b-tight. 2-reconfigurability of C is established by Lemma 4. We generalize the approach from [4, Lemma 1] to ab-constrained subgraphs to obtain a characterization of 1-reconfigurable b-tight even cycles in the case that M and N are maximum and no vertex of G is ab-tight. The proof is analogous to that of [4, Lemma 1] and is not given here. We denote by $\textsc{NotA}(M) := \{v \in V(G) \mid v \text{ is not } a\text{-tight in } M\}$ and $\textsc{NotB}(M) := \{v \in V(G) \mid v \text{ is not } b\text{-tight in } M\}$ the sets of vertices that are not a-tight in M and not b-tight in M, respectively.

$$\begin{aligned}
\textsc{Even}(M) \quad &:= \{ \quad v \in \textsc{NotA}(M) \mid \text{There is some even-length } M\text{-alternating} \\
&\qquad\qquad\qquad\qquad\qquad vw\text{-trail starting with an } M\text{-edge} \\
&\qquad\qquad\qquad\qquad\qquad \text{s.t. } w \in \textsc{NotB}(M)\} \\
\textsc{NotB}(G) \quad &:= \{ \quad v \in V(G) \mid \text{There is some maximum } M \subseteq G \text{ satisfying} \\
&\qquad\qquad\qquad\qquad\qquad \text{the degree constraints s.t. } v \in \textsc{NotB}(M)\}
\end{aligned}$$

The following lemma is a generalization of [4, Lemma 2] to the ab-constrained subgraph setting.

Lemma 5. *If M is maximum then* $\textsc{Even}(M) = \textsc{NotB}(G)$.

Proof. First, let $v \in \textsc{Even}(M)$. Then there is some vw-alternating trail T such that $w \in \textsc{NotB}(M)$. Then v is not a-tight and w is not b-tight in M and therefore $M' = M \triangle T$ is a maximum and satisfies the degree constraints. Since v is not b-tight in M' we have $v \in \textsc{NotB}(G)$. Therefore, $\textsc{Even}(M) \subseteq \textsc{NotB}(G)$. In order to prove that $\textsc{NotB}(G) \subseteq \textsc{Even}(M)$ assume that $v \in \textsc{NotB}(G)$. If $v \in \textsc{NotB}(M)$ then it is also in $\textsc{NotB}(G)$. Otherwise, let $N \subseteq G$ a maximum degree-constrained subgraph such that $v \in \textsc{NotB}(N)$. Then v is b-tight in M, but not in N. Suppose for a contradiction that there is no even-length (M, N)-alternating vw-trail such that $w \in \textsc{NotB}(M)$. Then for any even-length vw-trail w is b-tight. But since M and N are maximum, in any maximal open (M, N)-alternating vw-trail w is not b-tight. Furthermore, T cannot be a cycle since $\delta_M(v) \neq \delta_N(v)$. □

Lemma 6. *If G contains no ab-fixed vertices, M and N are maximum, and C is b-tight then $C \cap M$ is 1-reconfigurable to $C \cap N$ if and only if there is a vertex v of C such that $v \in$* $\textsc{Even}(M)$*.*

Proof. We first show that if $v \in C \cap \textsc{Even}(M)$ then $C \cap M$ is 1-reconfigurable to $C \cap N$. If $v \in C \cap \textsc{Even}(M)$ then there is a M-alternating vw-trail T of even length starting with an M-edge such that w is not b-tight. In the order given by the trail there is an earliest edge e such that none of the successors of e in T are C-edges. We distinguish two cases: First, assume that e is an M-edge. Without loss of generality, let u_0 be the C-vertex incident to e. We obtain the M-alternating subtrail $T' = u_0 \text{ — } \ldots \text{ — } w$ and $M' = M \triangle T$ is maximum and satisfies the degree constraints. Further, since u_0 is b-tight in M and w is not b-tight M, $T' \cap M$ is 1-reconfigurable to $T' \cap M'$ by Lemma 1. Then u_0 is not b-tight in M' and $C \cap M$ is 1-reconfigurable to $C \cap M'$ by Lemma 1. Since u_0 is not b-tight and

w is not a-tight in $C \cap M'$, $T' \cap M'$ is 1-reconfigurable to $T' \cap M$ and thus we can undo the changes to M caused by the reconfiguration on T'.

In the second case we assume that e is a $(G - M)$-edge. Then there is a latest C-edge on T such that none the successors of e in T are C-edges. Without loss of generality, let $e = u_0 — u_1$ such that we obtain an M-alternating subtrail $T' = u_0 — u_1 — \ldots — w$. Then u_0 is not a-tight in $T' \cap M$, w is not b-tight in $T' \cap M$, and therefore $T' \cap M$ is internally 1-reconfigurable to $T' - M$ by Lemma 1. In the resulting subgraph, u_0 is not b-tight and w is b-tight. By using Lemma 1 again, we can reconfigure the remaining parts of C and undo the modifications to $M - C$ caused by the previous step, by considering the trail $w — \ldots — u_1 — u_2 — \ldots — u_t(= u_0)$. Hence, in both cases $C \cap M$ is 1-reconfigurable to $C \cap N$. Note that each time we invoke Lemma 1, we use the assumption that no vertex of G is ab-fixed.

We now show that $v \in C \cap \textsc{Even}(M)$ is a necessary condition for $C \cap M$ to be 1-reconfigurable to $C \cap N$. If no vertex of C is in $\textsc{Even}(M) = \textsc{NotB}(G)$ (Lemma 5), then each vertex v of C is essentially ab-fixed in the sense that no maximum ab-constrained subgraph exists such that v is not b-tight. Therefore $C \cap M$ cannot be 1-reconfigurable to $C \cap N$. □

We now characterize k-reconfigurable alternatingly ab-tight cycles in $M \triangle N$ assuming that G contains no ab-fixed vertices. Such cycles cannot occur in the matching reconfiguration setting since no vertex of a cycle is a-tight in this case. There is some conceptual similarity to the proof of Lemma 4, but for the purpose of proving Theorem 1 we cannot assume that M and N are both maximum. Therefore, we cannot rely on Lemma 5, which simplifies the problem of finding a maximum ab-constrained subgraph M' such that a certain vertex is not b-tight to checking for the existence of an alternating trail. Instead, we now check if there is some ab-constrained subgraph M' such that the tightness of the C-vertices in M' differs from their tightness in M. The existence of a suitable M' can be checked in polynomial time by constructing and solving suitable ab-DCS instances.

Lemma 7. *If C is alternatingly ab-tight and G contains no ab-fixed vertices then $C \cap M$ is k-reconfigurable to $C \cap N$ for any $k \geq 1$ if and only if there is some ab-constrained $M' \subseteq G$ such that $C \cap M = C \cap M'$ and C is not alternatingly ab-tight in M'.*

Proof. Let us first assume that there is some ab-constrained $M' \subseteq G$ such that $C \cap M = C \cap M'$ and C is not alternatingly ab-tight in M'. We show that $C \cap M$ is 1-reconfigurable to $C \cap N$. Since C is not alternatingly ab-tight in M' there is some vertex v of C that is b-tight in M but not in M', or there is some vertex u of C that is a-tight in M but not in M'. We will consider in detail the case that there is some v of C that is b-tight in M but not in M'. The other case is analogous. Since $\delta_M(v) < \delta_{M'}(v)$ there is an open (M, M')-alternating vw-trail T in $M \triangle M'$ starting at v such that T cannot be extended at w. Without loss of generality, we assume that $v = u_0$ and $u_0 — u_1$ is a $(C - M)$-edge.

We consider two subcases: T has either even or odd length. First assume that T is even. Let $R = w — \ldots — u = u_0 — u_1 — \ldots — u_t$. Since T is even, $v = u_t$ is not a-tight in M and w is not b-tight in M. Furthermore, $R \triangle M$ satisfies the degree constraints. By Lemma 1, $R \cap M$ is 1-reconfigurable to $R \cap (R \triangle M) = ((C \cap N) + (T \cap M'))$. As a result, $C \cap M$ has been reconfigured to $C \cap N$. We now need to undo the changes in $T = R - C$ caused by the previous reconfiguration. Observe that v is not b-tight and w is not a-tight in $T \cap M'$. Therefore, we can invoke Lemma 1 again to reconfigure $T \cap M'$ to $T \cap M$. In the second subcase we assume that T is odd and let $R = w — \ldots — u = u_0 — u_1 — \ldots — u_{t-1}$ and let $S = u_{t-1} — u_0 — \ldots — w$. Then R and S are open and have even length, w is not a-tight and u_{t-1} is not b-tight in M. Therefore, by Lemma 1, $R \cap M$ is 1-reconfigurable to $R \cap (R \triangle M)$. Now u_{t-1} is not a-tight and w is not b-tight in $R \triangle M$. Therefore we can use Lemma 1 again in order to reconfigure $S \cap (R \triangle M)$ to $S \cap (S \triangle (R \triangle M)) = S \cap (M - e)$. As a result $C \cap M$ has been reconfigured to $C \cap N$.

In order to prove the converse statement, assume that there is no ab-constrained $M' \subseteq G$ such that $C \cap M = C \cap M'$ and C is not alternatingly ab-tight in M'. Then each vertex of C is essentially ab-fixed. Therefore, $C \cap M$ is not k-reconfigurable to $C \cap N$ for any $k \geq 1$. □

3.4 Reconfiguring ab-constrained Subgraphs

For the overall task of deciding if M is reconfigurable to N we will iteratively partition $M \triangle N$ into alternating trails as shown in Algorithm 2. Given M and N such that $M \triangle N$ is non-empty, the algorithm outputs a decomposition of $M \triangle N$ into trails $T_0, \ldots, T_{i-1}$ and a list of ab-constrained subgraphs $M_0, \ldots, M_{i-1}$ for some $i \geq 0$ such $M = M_0$ and $N = M_{i-1}$ such that for each j, $1 \leq j < i$,

$M_{j+1} = M_j \triangle T_j$. In each iteration i we need to find an M-augmenting uv-trail in $M_i \triangle N$ such that u and v are not b-tight in M_i. Since M_i is ab-constrained we can use for example the technique from [5, Section 2] to reduce the problem to obtaining an alternating path in an auxiliary graph that is constructed from $M \triangle N$. This approach produces a suitable M-augmenting trail if it exists in polynomial time. Since the number of iterations performed by the algorithm is bounded by $|E(M \triangle N)|$, the overall running time is polynomial in the size of the input graph. We are now ready to prove the main theorem. The structure of the proof is somewhat similar to the proof that matching reconfiguration can be solved in polynomial time, see [4, Proposition 2].

Algorithm 2: AlternatingTrailDecomposition

```
input  : $ab$-constrained $M, N \subseteq G$ s.t. $M \triangle N$ non-empty
output: Lists of alternating trails and $ab$-constrained subgraphs
$i \longleftarrow 0$;      $M_0 \longleftarrow M$
while $M_i \neq N$ do
    Find $M$-augmenting $uv$-trail $T$ in $M_i \triangle N$ s.t. $u$ and $v$ are not $b$-tight in $M_i$
    if such $T$ does not exist then
        Let $T$ be any maximal $(M_i, N)$-alternating trail in $M_i \triangle N$
    end
    $T_i \longleftarrow T$
    $M_{i+1} \longleftarrow M_i \triangle T$
    $i \longleftarrow i + 1$
end
return $[M_0, \ldots, M_{i-1}], [T_0, \ldots, T_{i-1}]$
```

Theorem 1. *st-DCSConn can be solved in polynomial time.*

Proof. Let $\mathcal{I} = (G', M', N', a', b', k)$ be a st-DCSConn instance. Let $F \subseteq G'$ be the M'-fixed subgraph produced by Algorithm 1. If $(M' \triangle N') \cap F$ is non-empty then some M'-fixed edge needs to be reconfigured, which is impossible by Proposition 1. Otherwise, we consider the subinstance $\mathcal{I}_{G'-F} = (G, M, N, a, b, k)$ (see Proposition 2).

Without loss of generality we assume that $|E(M)| \leq |E(N)|$. We process the alternating trails in $M \triangle N$ output by Algorithm 2 one by one in the given order. During the process, we observe the following types of (M, N)-alternating trails: i) even-length trails that are not b-tight or alternatingly ab-tight cycles, ii) M-augmenting trails, iii) N-augmenting trails, iv) b-tight even-length cycles, and v) alternatingly ab-tight even-length cycles. Since $|E(M)| \leq |E(N)|$ there are at least as many type ii)-trails as type iii)-trails. Note that each condition in the lemmas 1– 7 can be checked in polynomial time. We distinguish the following cases:

Case $k \geq 2$. By construction, in each step i, if T_i is in categories i)–iv) then $T \cap M_i$ is 2-reconfigurable to $T \cap M_{i+1}$ by Lemmas 2, 3, and 4. If T_i is an even-length alternatingly ab-tight cycle (type v)) then Lemma 7 gives necessary and sufficient conditions under which $T_i \cap M_i$ is k-reconfigurable to $T_i \cap M_{i+1}$ for any $k \geq 1$. These conditions can be checked in polynomial time and do not depend on what edges are present in M_i outside of T_i The reconfigurability of T_i is a property solely of T_i, G, and the degree bounds.

Case $k = 1$ and $|E(M)| < |E(N)|$. Trails of types i) and ii) are 1-reconfigurable by Lemmas 2 and 3. Due to the preference given to M-augmenting trails in Algorithm 2, if T_i is an N-augmenting trail or a b-tight cycle then $|E(M_i)| \geq |E(N)|$. Therefore by Lemma 3 or 6 we have that $T_i \cap M_i$ is 2-reconfigurable to M_{i+1}, but no intermediate subgraph is of size less than $|E(M)| - 1$. The alternatingly ab-tight cycles can be dealt with just as in the previous case.

Case $k = 1$ and $|E(M)| = |E(N)|$, both not maximum. In this case we increase the size of N by one, using an N-augmenting T, to obtain N'. We first reconfigure M to N' as in the case before. If successful, the result N' is 2-reconfigurable to N by Lemma 3. No intermediate subgraph is of size less than $|E(M)| - 1$.

Case $k = 1$, $|E(M)| = |E(N)|$, both maximum. Since M and N are maximum, each trail T_i is of type i), iv), or v). Therefore, each open trail is 1-reconfigurable by Lemma 2. We need to check the 1-reconfigurability of each cycle according to lemmas 6 (type iv)) and 7 (type v)). □

The running time of the decision procedure is dominated by the time needed to check the conditions of lemmas 6 and 7. Overall, this amounts to solving $O(|V(G)| \cdot |E(G)|^2)$ ab-DCS instances, which takes time $O(|E(G)|^{\frac{3}{2}})$ per instance using the algorithm from [3].

References

[1] Paul S. Bonsma. The complexity of rerouting shortest paths. *Theoretical Computer Science*, 510:1–12, 2013.

[2] Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *Journal of Graph Theory*, 67(1):69–82, 2011.

[3] Harold N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 448–456, New York, NY, USA, 1983. ACM.

[4] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(1214):1054–1065, 2011.

[5] Yossi Shiloach. Another look at the degree constrained subgraph problem. *Information Processing Letters*, 12(2):89–92, 1981.

[6] Jan van den Heuvel. The complexity of change. In Simon R Blackburn, Stefanie Gerke, and Mark Wildon, editors, *Surveys in Combinatorics 2013*, volume 409. London Mathematical Society Lectures Note Series, 2013.

[7] Marcin Wrochna. Homomorphism reconfiguration via homotopy. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 730–742, 2015.